REMARKS/ARGUMENTS

Claims 1-15 and 17-61 are pending in the application, claims 25-36 and 55-61 are withdrawn, and claims 1, 3, and 45 are amended. As discussed below, all of the claims are in condition for allowance. **But if after considering this response the Examiner does not allow all of the claims, then the Applicants' attorney requests that the Examiner contact him to schedule and conduct a telephone interview before issuing a subsequent Office Action.**

The Phone Interview With The Examiner On May 13, 2010 and Mr. Rapp's Declaration Under 37 C.F.R. § 1.132

The Applicants' attorney thanks the Examiner for participating in a telephone interview with the Applicants' attorney and coinventor Mr. John Rapp on May 13, 2010.

During the interview, the Applicants' attorney, Mr. Rapp, and the Examiner discussed the teachings of the Dretzka and Britton and why Dretzka and Britton, viewed alone or in combination, do not render the claims unpatentable. The Examiner agreed in principle that Dretzka and Britton, viewed alone or in combination, do not render the claims unpatentable, but requested that Mr. Rapp submit a declaration under 37 C.F.R § 1.132 explaining the teachings of Dretzka and Britton, and reiterating the points that Mr. Rapp made during the interview as to why Dretzka and Britton do not render the claims unpatentable. Therefore, Mr. Rapp's 1.132 declaration accompanies this response. And regarding claim 1, the examiner stated that he could read "data-processing units" on any part of a data-processing system, even on "a wire." Mr. Rapp and the Applicants' attorney stated that they did not agree with such an expansive interpretation, but indicated that they would try to formulate a term other than "data-processing units" to indicate the items with which the first and second parallel buffers are respectively associated.

Objections To The Specification

The Applicants' attorney continues to believe that the amended title of the application submitted in a response filed 19 December 2007 is descriptive of the claimed invention. But the Applicants' attorney will contemplate an amended title, and requests the Examiner's patience while he does so.

Rejection Of Claims 10-12, 14-15, 17-18, 37, 39-43, 45, and 47-49 Under 35 U.S.C. § 102(b) As Being Anticipated By U.S. Patent 4,703,475 To Dretzka

Claim 10

Claim 10 as amended recites a processor operable to generate data that includes only non-data-destination information, retrieve the generated data, load the retrieved data into a buffer, unload the data from the buffer, and process the unloaded data such that the processed data includes only non-data-destination information.

For example, referring, *e.g.*, to FIGS. 3-5 and paragraph [82] of the patent application, in an embodiment, a processor 42 is operable to generate data under the control of a first thread 100_3 of an application 80, the generated data including only non-data-destination information. The processor 42 is further operable to load the data into a buffer 106_5 under the control of a data-transfer object 86_{5a}, unload the data from the buffer 106_5 under the control of a second data-transfer object 86_{5b}, and process the unloaded data under the control of a second thread 100_4 of the same application 80 such that the processed data includes only non-data-destination information.

In contrast, Dretzka does not disclose a processor operable to generate data that includes only non-data-destination information, retrieve the generated data and load the retrieved data into a buffer, unload the data from the buffer, and process the unloaded data such that the processed data includes only non-data-destination information.

In the response to the final Office Action mailed November 17, 2009, Applicants' attorney argued that Dreztka discloses a buffer 120-4 (FIG. 5) and an input list 230-4 (FIG. 6) that store data packets each having a 3-byte packet header (FIG. 17) that

includes data-destination information that indicates the destination (*e.g.*, the logical channel LCN) for the data packet (*e.g.*, col. 7, lines 35-38 and lines 45-53, col. 8, line 55 – col. 9, line 12, and col. 9, line 60 – col. 10, line 11).

The Examiner disagreed with the Applicants' attorney's argument, however.

But as discussed in paragraphs [38], [62], [65], [66], and [67] of the accompanying 1.132 Declaration, even assuming that Dretzka's 3-byte packet header does not include data-destination information, Dretzka's data packets still include data-destination information, unlike the data recited in claim 10.

Dretzka discusses in detail only data transfers between two switching modules (*e.g.*, switching modules 10 and 20 per FIG. 2 of Dretzka). Dretzka's switching modules 10, 20, and 30 are compliant with the ISO OSI 7-layer model (*see*, *e.g.*, col. 1, lines 20-25 and the level designations in FIG. 4), and Dretzka discusses his layers 2-4 of this model in detail (*e.g.*, FIG. 4). The only functions that Dretzka attributes to his layers 2-4 in the transmitting module are functions for sending data packets over multiple physical links 40 to a receiving module (the layers 2-4 have additional functions to be compatible with the 7-layer model, but Dretzka is silent as to these functions). For example, referring to FIG. 2, the layers 2-4 of the module 10 operate to send data packets over links 40-0 – 40-4 to the module 20. And the only functions that Dretzka attributes to the layers 2-4 in the receiving module are functions for receiving the data packets over the multiple physical links 40 from the transmitting module. For example, referring to FIG. 2, the layers 2-4 of the module 20 receive data packets received over the links 40-0 – 40-4 from the module 10.

But referring to FIG. 1 of Dretzka, Dretzka's system includes at least three modules 10, 20, and 30.

Consequently, as discussed in paragraphs [38], [62], [65], [66], and [67] of the 1.132 Declaration, although Dretzka provides the details of only a data transfer between two modules, it is inherent that Dretzka's transmitting module (*e.g.*, the module 10) must include with the data an information header indicating the destination module (*e.g.*, the module 20 or the module 30) of the data, because without this destination header, the

layers 2-4 of the transmitting module would not “know” over which set of physical links (*e.g.*, the set 40-0 – 40-4 for the module 20 or the other links 40-5 – 40-m for the module 30) to send the data, and the receiving module would have no way of confirming whether it is an intended recipient of the received data.

In summary, even if the Examiner is correct that Dretzka’s 3-byte packet header (FIG. 17) is not destination data, it is at least inherent that Dretzka’s data packets still include destination data that indicates to which module 20 or 30 the module 10 is sending the data. Therefore, Dretzka at least fails to teach a processor operable to generate data that includes only non-data-destination information as recited in claim 10.

Claims 11-12, 14-15, and 17-18

These claims are patentable by virtue of their respective dependencies from claim 10.

Claim 37

Claim 37 recites loading data published with an application into a first buffer, the loaded data consisting only of non-data-destination information, and retrieving the published data from the buffer, the retrieved data consisting only of non-data-destination information.

For example, referring, *e.g.*, to FIGS 3-5 and paragraphs [66] – [70] of the patent application, in an embodiment a thread 100_1 of an application 80 publishes data that consists only of non-data-destination information, and a data-transfer object 86_{1a} loads the published data into a buffer 106_1, the loaded data consisting only of non-data-destination information. Then another data-transfer object 86_{1b} retrieves the published data from the buffer, the retrieved data consisting only of non-data-destination information. Next, because the channel 104_1 corresponds to one or more specified destinations, the data-transfer object 86_{1b} adds to the published data information (*e.g.*, a header) indicating the destination(s) of the published data. This relieves the application

80 and the thread 100_1 of the burden of adding destination information to the published data.

In contrast, Dretzka does not disclose loading data published with an application into a first buffer, the loaded data consisting only of non-data-destination information, and retrieving the published data from the buffer, the retrieved data consisting only of non-data-destination information.

In the response to the final Office Action mailed November 17, 2009, Applicants' attorney argued that Dreztka discloses a buffer 120-4 (FIG. 5) and an input list 230-4 (FIG. 6) that store data packets each having a 3-byte packet header (FIG. 17) that includes data-destination information that indicates the destination (*e.g.*, the logical channel LCN) for the data packet (*e.g.*, col. 7, lines 35-38 and lines 45-53, col. 8, line 55 – col. 9, line 12, and col. 9, line 60 – col. 10, line 11).

The Examiner disagreed with the Applicants' attorney's argument, however.

But as discussed in paragraphs [38], [62], [65], [66], and [67] of the accompanying 1.132 Declaration, even assuming that Dretzka's 3-byte packet header does not include data-destination information, Dretzka's data packets still include data-destination information, unlike the loaded and retrieved data recited in claim 37.

Dretzka discusses in detail only data transfers between two switching modules (*e.g.*, switching modules 10 and 20 per FIG. 2 of Dretzka). Dretzka's switching modules 10, 20, and 30 are compliant with the ISO OSI 7-layer model (*see*, *e.g.*, col. 1, lines 20-25 and the level designations in FIG. 4), and Dretzka discusses his layers 2-4 of this model in detail (*e.g.*, FIG. 4). The only functions that Dretzka attributes to his layers 2-4 in the transmitting module are functions for sending data packets over multiple physical links 40 to a receiving module (the layers 2-4 have additional functions to be compatible with the 7-layer model, but Dretzka is silent as to these functions). For example, referring to FIG. 2, the layers 2-4 of the module 10 operate to send data packets over links 40-0 – 40-4 to the module 20. And the only functions that Dretzka attributes to the layers 2-4 in the receiving module are functions for receiving the data packets over the multiple physical links 40 from the transmitting module. For example, referring to FIG.

2, the layers 2-4 of the module 20 receive data packets received over the links 40-0 – 40-4 from the module 10.

But referring to FIG. 1 of Dretzka, Dretzka's system includes at least three modules 10, 20, and 30.

Consequently, as discussed in paragraphs [38], [62], [65], [66], and [67] of the 1.132 Declaration, although Dretzka provides the details of only a data transfer between two modules, it is inherent that Dretzka's transmitting module (*e.g.*, the module 10) must include with the data an information header indicating the destination module (*e.g.*, the module 20 or the module 30) of the data, because without this destination header, the layers 2-4 of the transmitting module would not "know" over which set of physical links (*e.g.*, the set 40-0 – 40-4 for the module 20 or the other links 40-5 – 40-m for the module 30) to send the data, and the receiving module would have no way of confirming whether it is an intended recipient of the received data.

In summary, even if the Examiner is correct that Dretzka's 3-byte packet header (FIG. 17) is not destination data, it is at least inherent that Dretzka's data packets still include destination data that indicates to which module 20 or 30 the module 10 is sending the data. Therefore, Dretzka at least fails to teach loading and retrieving data consisting only of non-data-destination information as recited in claim 37.

Claims 39-43

These claims are patentable by virtue of their respective dependencies from claim 37.

Claim 45

Claim 45 recites receiving a message that includes data and that includes a message header that indicates a destination of the data, the destination corresponding to a software application, and loading into a buffer the received data without the message header, the buffer corresponding to the destination.

For example, referring, *e.g.*, to FIGS. 3-5 and paragraphs [76] – [77] of the patent application, in an embodiment a communication object 88 receives from a pipeline bus

50 a message that includes data and that includes a message header that indicates a destination of the data, the destinations being the software-application threads 100_1 and 100_2 (corresponding to a software application). A data-transfer object 86_{2b} strips the message header from the message and loads the data (without the stripped message header) into a buffer 106_2 that corresponds to the application threads 100_1 and 100_2, which are the destinations of the data.

In contrast, Dretzka does not disclose receiving a message that includes data and that includes a message header that indicates a destination of the data, the destination corresponding to a software application, and loading into a buffer the received data without the message header, the buffer corresponding to the destination.

As discussed in paragraphs [38], [62], [65], [66], and [67] of the accompanying 1.132 Declaration, the data packets disclosed in Dretzka inherently include a header indicating a destination switching module (*e.g.*, module 20 or module 30). But as also discussed in paragraphs [38] and [67], Dretzka does not discuss, and it is not inherent, how this header is handled. Therefore, Dretzka fails to disclose loading into a buffer received data without a message header that indicates a destination of the data, where the buffer corresponds to a destination software application for the data.

Claims 47-49

These claims are patentable by virtue of their respective dependencies from claim 45.

Rejection Of Claims 1-3 And 5-9 Under 35 U.S.C. § 103(a) As Being Obvious Over Dretzka In View Of U.S. 6,985,975 To Chamdani

Claim 1

Claim 1 as amended recites first and second parallel buffers respectively associated with first and second data-manipulation units, and a processor operable to load at least a portion of published data into the first buffer and to load at least the same portion of the published data into the second buffer.

For example, referring, *e.g.*, to FIGS. 3-5 and paragraphs [67] – [72] and [83] of the patent application, in an embodiment, a processor 42 is operable, under the control of an application thread 100_3, to publish data, and is operable, under the control of data-transfer object 86_{3a}, to load at least a portion of the published data into a first buffer 106_3, which is associated with a first data-manipulation unit (*e.g.*, a first hardwired pipeline 74) within the pipeline accelerator 44, and is operable, under the control of data-transfer object 86_{5a}, to load at least the same portion of the published data into a second buffer 106_5, which is associated with a second data-manipulation unit (*e.g.*, a second hardwired pipeline 74) within the pipeline accelerator and which is parallel to the first buffer 106_3.

In contrast, Dretzka does not include first and second parallel buffers respectively associated with first and second data-manipulation units, and a processor operable to load at least a portion of published data into the first buffer and to load at least the same portion of the published data into the second buffer as recited in claim 1. Referring to FIG. 5, even if the buffers, *e.g.*, 120-0 and 120-4, can be considered parallel and to be associated with different data-manipulation units, the processor 11 does not load the same data into these buffers.

And neither does Chamdani disclose first and second parallel buffers respectively associated with first and second data-manipulation units. In contrast, Chamdani's buffers (*e.g.*, FIFOs 102 and 103) are associated with a same data-manipulation unit (*e.g.*, whatever data-manipulation unit is connected to the single output of the coupler 110, and FIG. 5, step 408).

Consequently, the combination of Dretzka and Chamdani would at most suggest duplicating, for example, Dretzka's buffer 120-0, for redundancy, and loading these two buffers with the same data. But, unlike the first and second buffers recited in claim 1, the two buffers 120-0 would still be associated with only a single data-manipulation unit.

<u>Claims 2-3 and 5-9</u>

These claims are patentable by virtue of their dependencies from claim 1.

Rejection Of Claim 4 Under 35 U.S.C. § 103(a) As Being Obvious Over Dretzka In View Of Chamdani And Further In View Of The Examiner's Taking Of Official Notice

Claim 4

Claim 4 is patentable by virtue of its dependency from claim 1.

Furthermore, referring to paragraph [17] of the 1.132 Declaration, the application layer is layer 7.

And referring to, *e.g.*, paragraphs [32] and [33] of the 1.132 Declaration, Dretzka does not discuss layer 7.

Therefore, because Dretzka does not even discuss the application layer 7, there is nothing in Dretzka that would indicate it would be obvious to thread Dretzka's application.

Furthermore, referring to paragraphs [27] – [28] of the 1.132 Declaration, it would not be obvious to thread any of the functions discussed in Dretzka because such threading would reduce the overall data-transfer performance, such as by reducing the data-transfer rate and increasing the required message-buffering space. That is, Dretzka at least inherently teaches away from threading the layer stack because doing so would degrade, not improve, performance.

Consequently, at least because Dretzka teaches away from threading any of its described functions, the combination of Dretzka, Chamdani, and the Examiner's official notice fail to render claim 4 obvious.

Rejection Of Claims 13-14, 38, 44, 46, And 50 Under 35 U.S.C. § 103(a) As Being Obvious Over Dretzka In View Of The Examiner's Taking Of Official Notice

The Applicants' attorney maintains his objection to the Examiner's taking of official notice for these claims as discussed in at least the response to the Final Office Action mailed November 17, 2009

Claim 13

Claim 13 is patentable by virtue of its dependency from claim 10, and for reasons similar to those recited above in support of the patentability of claim 4.

Claim 14

Claim 14 is patentable by virtue of its dependency from claim 10.

Claim 38

Claim 38 is patentable by virtue of its dependency from claim 37, and for reasons similar to those discussed above in support of the patentability of claim 4.

Claim 44

Claim 44 as amended is patentable by virtue of its dependency from claim 37.

Claim 46

Claim 46 is patentable by virtue of its dependency from claim 45, and for reasons similar to those discussed above in support of the patentability of claim 4.

Claim 50

Claim 50 is patentable by virtue of its dependency from claim 45.

Rejection Of Claims 19-24 and 51-54 Under 35 U.S.C. § 103(a) As Being Obvious Over Dretzka In View Of U.S. Patent 6,216,191 to Britton

Claim 19

Claim 19 recites a processor operable to execute an application and first and second data-transfer objects, to publish data under the control of the application, to load the published data into a buffer under the control of the first data-transfer object, to retrieve the published data from the buffer under the control of the second data-transfer object, to construct a message under the control of the second data-transfer object where the message includes the retrieved published data and information indicating a destination of the retrieved published data, and to drive the message onto a bus under

the control of the communication object, and a pipeline accelerator that includes the destination of data and that is operable to receive the message, to recover the data from the message, and to process the recovered data at the destination without executing a program instruction.

For example, referring, *e.g.,* to FIGS. 3-5 and paragraphs [67] – [72] of the patent application, in an embodiment, a processor 42 is operable to execute an application 80 and first and second data-transfer objects 86_{1a} and 86_{1b}, to publish data under the control of the application, to load the published data into a buffer 106_1 under the control of the first data-transfer object 86_{1a}, to retrieve the published data from the buffer under the control of the second data-transfer object, to construct a message that includes data and information indicating a destination of the data within a pipeline accelerator 44, and to drive the message onto a bus 50. The accelerator 44 is operable to receive the message from the bus 50, to recover the data from the message, and to process the recovered data at the destination without executing a program instruction.

In contrast, neither Dreztka nor Britton, viewed alone or in combination, would have taught, suggested to, or motivated one of skill in the art to modify Britton's FPGA interface 100 and processor 102 so that Britton's processor and FPGA 104 could have communicated with each other via messages per claim 19.

First, as discussed in paragraphs [38], [62], [65], [66], and [67] of the 1.132 declaration, the examiner has failed to make a prima facie case of obviousness because Dretzka does not teach or suggest limitations of claim 19 that the examiner erroneously stated are found in Dretzka. Although it is inherent that Dretzka's data packets include headers that indicate a destination (*e.g.*, switching module 20 or module 30) of the data, Dretzka is silent as to how these headers are generated or added to the data, or are otherwise handled; therefore, Dretzka at least fails to disclose or suggest the buffer and data-transfer objects recited in claim 19.

Second, as discussed in paragraphs [86] – [95] of the 1.132 declaration, Dretzka does not contain information sufficient to allow one of ordinary skill in the art to modify Britton's FPGA interface 100 and processor 102 to communicate via messages. To so modify Britton's FPGA interface 100 and processor 102 in view of Dretzka would require

constructing both the FPGA interface and the processor according to the ISO OSI 7-layer model taught by Dretzka. But as discussed in paragraphs [86] – [95] of the 1.132 declaration, the teachings of Dretzka are insufficient for one of ordinary skill in the art to construct Britton's FPGA interface 100 and processor 102 according to the ISO OSI 7-layer model. Dretzka discusses only layers 2-4 of the 7-layer model, and for these layers only discusses some of the functions that they perform. Consequently, Dretzka provides none of the information that one of skill in the art would have needed to construct layers 1 and 5-7 of Britton's processor 102, and at best provides only some of the information that one of skill in the art would have needed to construct layers 2-4 of Britton's processor 102. And for these same reasons and further because Dretzka's switching modules 10, 20, and 30 are only processor based and are not FPGA based, Dretzka provides absolutely no information that one of skill in the art would have needed to construct layers 1-7 for Britton's FPGA interface 100.

Third, still referring to paragraphs [86] – [95] of the 1.132 declaration, neither does Briton contain information sufficient to allow one of ordinary skill in the art to modify Britton's FPGA interface 100 and processor 102 to communicate via messages.

And fourth, as at least can be inferred from paragraphs [86] – [95] of the 1.132 declaration, Dretzka at least inherently teaches away from modifying Britton's FPGA interface 100 and processor 102 to communicate via messages. Referring to Dretka's FIGS. 5-6, to be compliant with the ISO OSI 7-layer standard, the 7 layers in a transmitting module must be symmetrical to the 7 layers in the receiving module, and vice versa. That is, for example, if a layer in the transmitting module encrypts the data according to a particular algorithm, then the same layer in the receiving module must decrypt the data according to the same algorithm. Furthermore, still referring to FIGS. 5-6, to implement the 7 layers in the module 10 is relatively complex. But referring to FIGS. 1, 5, and 6, because all modules (*e.g.*, modules 10, 20, and 30) are the same, then once the module design and debug is complete, it can be replicated to easily populate a system with as many modules as desired. Furthermore, the identical maintenance (*e.g.*, software updates) can be performed on all of the modules. But including in the system processor-based modules such as the modules 10, 20, and 30, and also including in the system FPGA-based modules such as a modified version of

Britton's FPGA 104, would require the design, debug, and maintenance of two module types for the system. Furthermore, Dretzka gives no indication as to whether his system will benefit from replacing some of the processor-based modules (*e.g.*, 10, 20, and 30) with FPGA-based modules. Therefore, even if one of skill in the art could determine how to modify Briton's FPGA 100 to communicate with Dretzka's processor-based switching modules 10, 20, and 30, because replacing some of the processor modules with FPGA modules would significantly increase the cost and effort to design, debug, and maintain Dretzka's system with no offsetting benefit, Dretzka inherently teaches away from replacing any of its processor-based modules with such modified ones of Briton's FPGA-based modules.

Claims 20-21

These claims are patentable by virtue of their dependencies from claim 19.

Claim 22

Claim 22 recites a pipeline accelerator operable to generate data without executing a program instruction, to generate a header including information indicating a destination of the data, and to package the data and header into a message, and a processor operable to receive the message under the control of a communication object, to load into a buffer under the control of a first data-transfer object the received data without the header, the buffer corresponding to the destination of the data, to unload the data from the buffer under the control of a second data-transfer object, and to process the unloaded data under the control of an application.

For example, referring, *e.g.*, to FIGS. 3 - 5, paragraph [49], and paragraph [74] of the patent application, a pipeline accelerator 44 is operable to generate data without executing a program instruction, to generate a header including information indicating a destination of the data (*e.g.*, a thread 100 of an application program 80 of FIG. 5), to package the data and header into a message, and to drive the message onto a bus 50. And a processor 42 is operable to execute an application 80, first and second data-transfer objects 86_{4a} and 86_{4b}, and a communication object 88, to receive the message under the control of the communication object, to load into a buffer 106_4 under the control of the first data-transfer object the received data without the header, to

unload the data from the buffer under the control of the second data-transfer object, and to process the unloaded data under the control of the application.

In contrast, neither Dreztka nor Britton, viewed alone or in combination, would have taught, suggested to, or motivated one of skill in the art to modify Britton's FPGA interface 100 and processor 102 so that Britton's processor and FPGA 104 could have communicated with each other via messages per claim 22.

First, as discussed in paragraphs [38], [62], [65], [66], and [67] of the 1.132 declaration, the examiner has failed to make a prima facie case of obviousness because Dretzka does not teach or suggest limitations of claim 22 that the examiner erroneously stated are found in Dretzka. Although it is inherent that Dretzka's data packets include headers that indicate a destination (*e.g.*, switching module 20 or module 30) of the data, Dretzka is silent as to how these headers are stripped from the data, or are otherwise handled; therefore, Dretzka at least fails to disclose or suggest the buffer and data-transfer objects recited in claim 22.

Second, as discussed in paragraphs [86] – [95] of the 1.132 declaration, Dretzka does not contain information sufficient to allow one of ordinary skill in the art to modify Britton's FPGA interface 100 and processor 102 to communicate via messages. To so modify Britton's FPGA interface 100 and processor 102 in view of Dretzka would require constructing both the FPGA interface and the processor according to the ISO OSI 7-layer model taught by Dretzka. But as discussed in paragraphs [86] – [95]of the 1.132 declaration, the teachings of Dretzka are insufficient for one of ordinary skill in the art to construct Britton's FPGA interface 100 and processor 102 according to the ISO OSI 7-layer model. Dretzka discusses only layers 2-4 of the 7-layer model, and for these layers only discusses some of the functions that they perform. Consequently, Dretzka provides none of the information that one of skill in the art would have needed to construct layers 1 and 5-7 of Britton's processor 102, and at best provides only some of the information that one of skill in the art would have needed to construct layers 2-4 of Britton's processor 102. And for these same reasons and further because Dretzka's switching modules 10, 20, and 30 are only processor based and are not FPGA based, Dretzka provides absolutely no information that one of skill in the art would have needed to construct layers 1-7 for Britton's FPGA interface 100.

Third, still referring to paragraphs [86] – [95] of the 1.132 declaration, neither does Briton contain information sufficient to allow one of ordinary skill in the art to modify Britton's FPGA interface 100 and processor 102 to communicate via messages.

And fourth, as at least can be inferred from paragraphs [86] – [95] of the 1.132 declaration, Dretzka at least inherently teaches away from modifying Britton's FPGA interface 100 and processor 102 to communicate via messages. Referring to Dretka's FIGS. 5-6, to be compliant with the ISO OSI 7-layer standard, the 7 layers in a transmitting module must be symmetrical to the 7 layers in the receiving module, and vice versa. That is, for example, if a layer in the transmitting module encrypts the data according to a particular algorithm, then the same layer in the receiving module must decrypt the data according to the same algorithm. Furthermore, still referring to FIGS. 5-6, to implement the 7 layers in the module 10 is relatively complex. But referring to FIGS. 1, 5, and 6, because all modules (*e.g.*, modules 10, 20, and 30) are the same, then once the module design and debug is complete, it can be replicated to easily populate a system with as many modules as desired. Furthermore, the identical maintenance (*e.g.*, software updates) can be performed on all of the modules. But including in the system processor-based modules such as the modules 10, 20, and 30, and also including in the system FPGA-based modules such as a modified version of Britton's FPGA 104, would require the design, debug, and maintenance of two module types for the system. Furthermore, Dretzka gives no indication as to whether his system will benefit from replacing some of the processor-based modules (*e.g.*, 10, 20, and 30) with FPGA-based modules. Therefore, even if one of skill in the art could determine how to modify Briton's FPGA 100 to communicate with Dretzka's processor-based switching modules 10, 20, and 30, because replacing some of the processor modules with FPGA modules would significantly increase the cost and effort to design, debug, and maintain Dretzka's system with no offsetting benefit, Dretzka inherently teaches away from replacing any of its processor-based modules with such modified ones of Briton's FPGA-based modules.

Claims 23-24

These claims are patentable by virtue of their dependencies from claim 22.

Claim 51

Claim 51 recites publishing data with an application running on a processor, loading the published data into a buffer with a first data-transfer object running on the processor, retrieving the published data from the buffer with a second data-transfer object running on the processor, generating information that indicates a hardwired pipeline for processing the retrieved data, packaging the retrieved data and the information into a message, driving the message onto a bus with a communication object running on the processor, and receiving the message from the bus and processing the published data with the indicated hardwired pipeline, which is part of a pipeline accelerator that includes a field-programmable gate array.

In contrast, neither Dreztka nor Britton, viewed alone or in combination, would have taught, suggested to, or motivated one of skill in the art to modify Britton's FPGA interface 100 and processor 102 so that Britton's processor and FPGA 104 could have communicated with each other via messages per claim 51.

First, as discussed in paragraphs [38], [62], [65], [66], and [67] of the 1.132 declaration, the examiner has failed to make a prima facie case of obviousness because Dretzka does not teach or suggest limitations of claim 51 that the examiner erroneously stated are found in Dretzka. Although it is inherent that Dretzka's data packets include headers that indicate a destination (*e.g.*, switching module 20 or module 30) of the data, Dretzka is silent as to how these headers are generated or added to the data, or are otherwise handled; therefore, Dretzka at least fails to disclose or suggest the loading, retrieving, generating, and packaging steps recited in claim 51.

Second, as discussed in paragraphs [86] – [95] of the 1.132 declaration, Dretzka does not contain information sufficient to allow one of ordinary skill in the art to modify Britton's FPGA interface 100 and processor 102 to communicate via messages. To so modify Britton's FPGA interface 100 and processor 102 in view of Dretzka would require constructing both the FPGA interface and the processor according to the ISO OSI 7-layer model taught by Dretzka. But as discussed in paragraphs [86] – [95]of the 1.132 declaration, the teachings of Dretzka are insufficient for one of ordinary skill in the art to construct Britton's FPGA interface 100 and processor 102 according to the ISO OSI

7-layer model. Dretzka discusses only layers 2-4 of the 7-layer model, and for these layers only discusses some of the functions that they perform. Consequently, Dretzka provides none of the information that one of skill in the art would have needed to construct layers 1 and 5-7 of Britton's processor 102, and at best provides only some of the information that one of skill in the art would have needed to construct layers 2-4 of Britton's processor 102. And for these same reasons and further because Dretzka's switching modules 10, 20, and 30 are only processor based and are not FPGA based, Dretzka provides absolutely no information that one of skill in the art would have needed to construct layers 1-7 for Britton's FPGA interface 100.

Third, still referring to paragraphs [86] – [95] of the 1.132 declaration, neither does Briton contain information sufficient to allow one of ordinary skill in the art to modify Britton's FPGA interface 100 and processor 102 to communicate via messages.

And fourth, as at least can be inferred from paragraphs [86] – [95] of the 1.132 declaration, Dretzka at least inherently teaches away from modifying Britton's FPGA interface 100 and processor 102 to communicate via messages. Referring to Dretka's FIGS. 5-6, to be compliant with the ISO OSI 7-layer standard, the 7 layers in a transmitting module must be symmetrical to the 7 layers in the receiving module, and vice versa. That is, for example, if a layer in the transmitting module encrypts the data according to a particular algorithm, then the same layer in the receiving module must decrypt the data according to the same algorithm. Furthermore, still referring to FIGS. 5-6, to implement the 7 layers in the module 10 is relatively complex. But referring to FIGS. 1, 5, and 6, because all modules (*e.g.*, modules 10, 20, and 30) are the same, then once the module design and debug is complete, it can be replicated to easily populate a system with as many modules as desired. Furthermore, the identical maintenance (*e.g.*, software updates) can be performed on all of the modules. But including in the system processor-based modules such as the modules 10, 20, and 30, and also including in the system FPGA-based modules such as a modified version of Britton's FPGA 104, would require the design, debug, and maintenance of two module types for the system. Furthermore, Dretzka gives no indication as to whether his system will benefit from replacing some of the processor-based modules (*e.g.*, 10, 20, and 30) with FPGA-based modules. Therefore, even if one of skill in the art could determine

how to modify Briton's FPGA 100 to communicate with Dretzka's processor-based switching modules 10, 20, and 30, because replacing some of the processor modules with FPGA modules would significantly increase the cost and effort to design, debug, and maintain Dretzka's system with no offsetting benefit, Dretzka inherently teaches away from replacing any of its processor-based modules with such modified ones of Briton's FPGA-based modules.

Claim 52

Claim 52 is patentable by virtue of its dependency from claim 51.

Claim 53

Claim 53 recites generating with a pipeline accelerator and without executing a program instruction a message header that includes a destination of data, generating with the pipeline accelerator and without executing a program instruction a message that includes the header and the data, driving the message onto a bus with the pipeline accelerator, receiving the message from the bus with a communication object running on a processor, loading into a buffer with a first data-transfer object running on the processor the received data absent the header, the buffer being identified by the destination, unloading the data from the buffer with a second data-transfer object running on the processor, and processing the unloaded data with the processor.

In contrast, neither Dreztka nor Britton, viewed alone or in combination, would have taught, suggested to, or motivated one of skill in the art to modify Britton's FPGA interface 100 and processor 102 so that Britton's processor and FPGA 104 could have communicated with each other via messages per claim 53.

First, as discussed in paragraphs [38], [62], [65], [66], and [67] of the 1.132 declaration, the examiner has failed to make a prima facie case of obviousness because Dretzka does not teach or suggest limitations of claim 53 that that the examiner erroneously stated are found in Dretzka. Although it is inherent that Dretzka's data packets include headers that indicate a destination (*e.g.*, switching module 20 or module 30) of the data, Dretzka is silent as to how these headers are stripped from the data, or are otherwise handled; therefore, Dretzka at least fails to disclose or suggest

the loading and unloading steps recited in claim 53.

Second, as discussed in paragraphs [86] – [95] of the 1.132 declaration, Dretzka does not contain information sufficient to allow one of ordinary skill in the art to modify Britton's FPGA interface 100 and processor 102 to communicate via messages. To so modify Britton's FPGA interface 100 and processor 102 in view of Dretzka would require constructing both the FPGA interface and the processor according to the ISO OSI 7-layer model taught by Dretzka. But as discussed in paragraphs [86] – [95] of the 1.132 declaration, the teachings of Dretzka are insufficient for one of ordinary skill in the art to construct Britton's FPGA interface 100 and processor 102 according to the ISO OSI 7-layer model. Dretzka discusses only layers 2-4 of the 7-layer model, and for these layers only discusses some of the functions that they perform. Consequently, Dretzka provides none of the information that one of skill in the art would have needed to construct layers 1 and 5-7 of Britton's processor 102, and at best provides only some of the information that one of skill in the art would have needed to construct layers 2-4 of Britton's processor 102. And for these same reasons and further because Dretzka's switching modules 10, 20, and 30 are only processor based and are not FPGA based, Dretzka provides absolutely no information that one of skill in the art would have needed to construct layers 1-7 for Britton's FPGA interface 100.

Third, still referring to paragraphs [86] – [95] of the 1.132 declaration, neither does Briton contain information sufficient to allow one of ordinary skill in the art to modify Britton's FPGA interface 100 and processor 102 to communicate via messages.

And fourth, as at least can be inferred from paragraphs [86] – [95] of the 1.132 declaration, Dretzka at least inherently teaches away from modifying Britton's FPGA interface 100 and processor 102 to communicate via messages. Referring to Dretka's FIGS. 5-6, to be compliant with the ISO OSI 7-layer standard, the 7 layers in a transmitting module must be symmetrical to the 7 layers in the receiving module, and vice versa. That is, for example, if a layer in the transmitting module encrypts the data according to a particular algorithm, then the same layer in the receiving module must decrypt the data according to the same algorithm. Furthermore, still referring to FIGS. 5-6, to implement the 7 layers in the module 10 is relatively complex. But referring to FIGS. 1, 5, and 6, because all modules (*e.g.*, modules 10, 20, and 30) are the same,

then once the module design and debug is complete, it can be replicated to easily populate a system with as many modules as desired. Furthermore, the identical maintenance (*e.g.*, software updates) can be performed on all of the modules. But including in the system processor-based modules such as the modules 10, 20, and 30, and also including in the system FPGA-based modules such as a modified version of Britton's FPGA 104, would require the design, debug, and maintenance of two module types for the system. Furthermore, Dretzka gives no indication as to whether his system will benefit from replacing some of the processor-based modules (*e.g.*, 10, 20, and 30) with FPGA-based modules. Therefore, even if one of skill in the art could determine how to modify Briton's FPGA 100 to communicate with Dretzka's processor-based switching modules 10, 20, and 30, because replacing some of the processor modules with FPGA modules would significantly increase the cost and effort to design, debug, and maintain Dretzka's system with no offsetting benefit, Dretzka inherently teaches away from replacing any of its processor-based modules with such modified ones of Briton's FPGA-based modules.

Claim 54

Claim 54 is patentable by virtue of its dependency from claim 53.

CONCLUSION

In view of the foregoing, claims 2, 4-15, 17-24, 37-43, 46-49, and 51-54 as previously pending, and claims 1, 3, and 45 as amended are in condition for allowance. Therefore, the issuance of a formal Notice of Allowance at an early date is respectfully requested. **If the Examiner does not agree that all claims are in condition for allowance, the Examiner is respectfully requested to telephone the undersigned prior to issuing an action rejecting the claims to schedule a telephone interview.**

In the event additional fees are due as a result of this amendment, you are hereby authorized to charge such payment to Deposit Account No. 07-1897.

DATED this 28th day of May 2010.

Respectfully submitted,

GRAYBEAL JACKSON LLP

/Bryan A. Santarelli/

Bryan A. Santarelli
Registration No. 37,560

Customer No. 00996

Graybeal Jackson LLP
400 - 108th Avenue NE, Suite 700
Bellevue, Washington 98004-5565
Telephone: 425.455.5575
Facsimile: 425.455.1046